Towards Easy Matching Between Statistical Linked Data: Dimension Patterns

Hideto Sato[1] and Wen Wen[1]

[1] Tokyo International University, Kawagoe, Japan
{sato,s11170003bb}@tiu.ac.jp

Abstract. Users of statistics expect linked data technology to make it easy to discover statistical data from different sources that can match each other. We carried out a trial matching between statistical linked data and pointed out the importance of upper concepts concerning dimensions and schema-level description about external linkages. We surveyed statistical linked data endpoints, finding that many dimension descriptions cannot have the information about the upper concepts and the external linkages. It is because they directly use external resources, to which local description cannot be added. Hence, we propose two patterns of dimension description to avoid this problem. So far, only small numbers of upper concepts have been supplied for statistical data. Therefore, the parts of the patterns concerning upper concepts are preparatory for the future and the benefit of the patterns is limited at present.

Keywords: statistical linked data, RDF data cube, dimension, discovering, matching, alignment, pattern, alternate class, adapter, upper concept

1 Introduction

Users of statistics often use data from different sources widely distributed. They tend to encounter difficulties in discovering data that can match each other. It is mainly because the terminology and the structure used for data are different from each source [1]. Statistical linked data are expected to improve this situation. Linked data create typed links between data from different sources and provides integrated access to data from a wide range of distributed and heterogeneous data sources [2].

Standardization of vocabulary for describing statistical linked data is now in progress. It is "The RDF Data Cube Vocabulary" (QB for short) by W3C [3]. Many statistics have already been published as linked data based on QB.

On the other hand, as to linked data in general, many methods have been proposed for matching heterogeneous data from different sources [4]. In order to discover appropriate datasets from widely distributed statistics, we consider that a promising way is rough selection of the candidates by using upper ontology and its refinement by using the scheme-level information.[1] As for upper ontology, QB provides a bridge to some upper concepts by referring to the SDMX-RDF vocabulary based on the SDMX

[1] The importance of upper concepts in a statistical data domain was pointed out in [5].

adfa, p. 1, 2011.

cross-domain concepts [6]. We use the upper-level resources in the SDMX-RDF vocabulary as examples of upper concepts in this paper.

For investigating the benefit of upper concepts and schema-level information in a matching process, we carried out a trial matching between linked data. We found out that, if dimensions were defined in a better way, upper-level resources were referred to properly and schema-level description about external links existed, the matching would be done in a considerably automatic way. Next, we surveyed sites where statistical linked data were published via SPARQL endpoints, investigating how they used upper concepts and external links.

Based on this trial and the survey, we propose patterns of dimension description. These patterns are basically according to the example found in the QB draft and enable to describe upper concepts and external link information in the schema level.

So far, only small numbers of upper concepts are available for statistical data, so that the benefit of the patterns proposed here is limited. However, with the growth of statistical linked data, upper concepts for them will be supplied by many communities. We think that the parts of the patterns concerning upper concepts are preparatory for the next stage of statistical linked data.

In addition, for instance matching, we deal with merely exact matching. There are many complex problems in matching statistical data, such as handling similarity or versioning of dimension values. Even in these cases, it is necessary to find target datasets and target dimensions before instance matching. We consider that discovering datasets and dimensions may be similar to our cases.

In Section 2, the above trial matching is explained and the findings in it are summarized. In Section 3, we outline the result of our survey of statistical linked data sites. In Section 4, based on the findings in the trial and the survey, we propose patterns each of which describes a dimension and its related resources. Finally, we summarize our conclusion and future works in Section 5.

Throughout this paper, we use prefixes such as qb:, sdmx-dimension:, sdmx-code:, sdmx-concept:, eg: and interval:. The URIs for these, please refer to the QB draft [3].

2 Trial Matching Between Statistics

2.1 Outline of the Trial

We carried out a trial matching in order to investigate what helps automatic matching between statistical data from different sources. This was done in September 2012. We took up Italian Immigration Statistics (ItImmStat for short) and World Bank Statistics (WBStat for short) as samples. We tried to get the numbers of immigrants to Italy by birth country by year from ItImmStat and the total population by country by year from WBStat. Then we integrated them.

Both statistics were published as linked data based on QB via SPARQL endpoints.[2]

[2] ItImmStat: `http://sparql.linkedopendata.it/istat`
WBStat: `http://worldbank.270a.info/`

We could get the integrated data by creating an application with SPARQL queries. However, it was difficult to formulate the above SPARQL queries mechanically, and we were forced to read the labels or the comments in order to understand the meaning of each dimension and to identify dimension values to be used for matching. It was because the two linked data were different from each other in description though both were based on QB.

In this trial, we extracted observation data by matching country codes of area dimensions of the two statistics and by matching year codes of time dimensions of them. The following subsections explain problems and solutions that we found in this trial for each dimension.

2.2 Matching Between Area Dimensions

Fig.1 illustrates an RDF graph related to the matching of country codes between ItImmStat and WBStat. Broken lines show resources/properties that did not appear in the linked data.

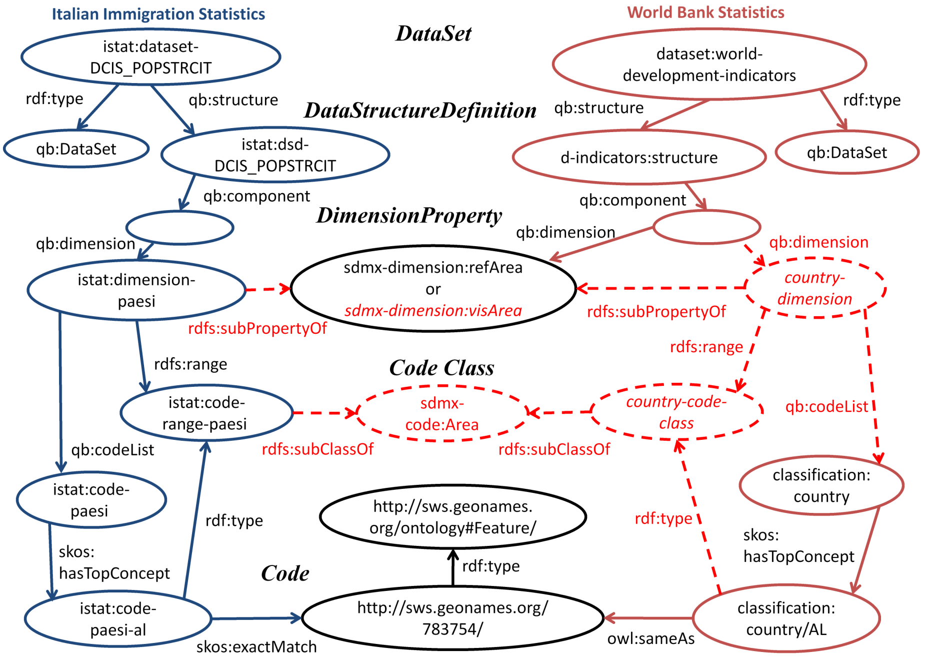


Fig. 1. Matching between area codes

At first, it is necessary to identify dimension properties for the matching. In this case, we tried to identify a dimension for places of residence in WBStat and a dimension for places of birth in ItImmStat. In SDMX, REF_AREA is a concept for the area directly related to the statistical phenomenon (e.g. places of residence) and VIS_AREA is a concept for the secondary area (e.g. places of birth) [6]. Hence, we looked for a dimension that was a subproperty of sdmx-dimension:refArea in WBStat and a dimension that was a subproperty of sdmx-dimension:visArea in ItImmStat.

We could find sdmx-dimension:refArea easily in WBStat because sdmx-dimension:refArea was directly used as a dimension property. However, we could not find the dimension for birthplace in ItImmStat mechanically. We were forced to read the labels and the comments to find istat:dimension-paesi as the birth country dimension. This is not a problem of ItImmStat. sdmx-dimension:visArea has been removed in the current version of sdmx-dimension.ttl while sdmx-concept:visArea exists in sdmx-concept.ttl. If there had existed upper-level resources to distinguish “place of birth” and “place of residence” and if they had been referred to as the super dimension properties, we could have done this step automatically.

After finding the dimension candidates, we have to check whether the candidates are appropriate to the purpose. In this case, it was whether the dimension values were country codes or not. As for area, there exist many types of division, such as countries, domestic administrative areas, and river basins. These are differences among code classes (range classes of dimensions). While the SDMS-RDF provides sdmx-code:Area that represents the uppermost concept of area codes, it does not supply subclasses of sdmx-code:Area. If there had existed upper-level resources that distinguished the types of area code, we could have done this step automatically, too.

Herein, we found one problem. Please, look at Fig.1. The dimension property of WBStat is the upper-level “sdmx-dimension:refArea”. This is an external resource, so that it is impossible to designate its range locally. Therefore, there is no way to declare a link to the upper-level code class. This problem can be resolved by introducing a local dimension property and a local code class as drawn with broken lines in Fig.1.

Next, we examined instance matching between area codes. For ItImmStat, we could easily get the area codes because the dimension property had a link to its code list resource. On the contrary, since WBStat used the upper-level dimension property as above, neither its range nor its code list were available in the dimension definition. We had to get the area codes by searching the observation data. We think that the reshaping of the dimension definition as mentioned above is desirable from this point of view, too.

Since both ItImmStat and WBStat defined the country codes as their local URIs, these codes themselves did not match each other. However, ItImmStat linked their country codes to those of Geonames and DBPedia by using skos:exactMatch and WBStat did the same by using owl:sameAs. By using them, the country codes of the two datasets could match. As might be expected, we could not know, in the schema level, whether links to external codes existed or not. We had to retrieve the code instances to know it. If there had existed schema level information about external links, we could have done the matching rather automatically.

2.3 Matching Between Time Dimensions

The description of time dimensions in the both linked data was similar to that of the area dimensions. SDMX defines REF_PERIOD as a concept for a period of time or point in time related to measured observations. ItImmStat defined the time dimension property locally and declared that it was a subproperty of sdmx-dimension:refPeriod. WBStat used sdmx-dimension:refPeriod directly as the time dimension property. Hence, it was easy to find time dimensions to match.

As the dimension property in ItImmStat had a link to its code list, we could easily get the time codes in use. In the case of WBStat, neither range nor code list was available. We had to search the observation data to get the time codes in use.

ItImmStat defined the time codes as their local URIs and linked them, by using skos:exactMatch, to those defined by data.gov.uk. WBStat used time the codes defined by data.gov.uk directly in the observation data. Hence, we could match them.

3 Survey of Statistical Linked Data Endpoints

We surveyed existing statistical linked data for checking how they describe dimensions and their related resources. We looked up the sites listed in the "Data Cube Implementations" page of W3C [3] and selected nine sites, each of which was accessible via SPARQL endpoints and published datasets having both area and time dimensions. Although the number of sites surveyed is very small and about half of the creators of the datasets are the same, we suppose that we can figure out approximate tendency of currently published statistical linked data.

3.1 Area Dimension

Table.1 summarizes the survey result about area dimensions. The "DSD" column shows a row id indicating a data structure definition. For the information about row ids, please look at the footnote below the table. The "dimension" column shows the type of a dimension property. Herein, the value "local" indicates a dimension property locally defined by the site. "sd:refArea" is an abbreviation of "sdmx-dimension:refArea".

The "generic dimension" column shows the rdfs:subPropertyOf value of the dimension property. The "range class" column shows the rdfs:range value of the dimension property, and "generic range class" is an rdfs:subClassOf value of it. The "code" column shows the type of code in use. The "alternate code" column shows the type of code given via skos:exactMatch or owl:sameAs. Herein, geonames, dbpedia and eurostat mean the codes defined by the respective organizations. This column lists only better-known ones if many external codes are linked.

Please look at the "dimension" and "generic dimension" columns. 5 DSDs in 12 ones use dimension properties defined locally and referring to the upper-level "sdmx-dimension:refArea". 3 DSDs directly use this upper-level dimension property. Hence, for two-thirds of DSDs, the area dimension is identifiable in a schema level. However, the latter three DSDs have the same problem as in WBStat in Section 2, having no information about their range classes.

[3] http://www.w3.org/2011/gld/wiki/Data_Cube_Implementations

For the range classes, many sites define them but they are almost all local, so that they give no information for matching. Only the row "c" (Bathing Water Quality) specifies the generic range class (WGS84: World Geodetic System). As for the range classes, various types such as countries, domestic administrative areas, river basins and geographic points were used. In order to judge the matching possibility precisely, upper concepts representing them are necessary.

Next, let us examine instance codes. While almost all datasets surveyed here use locally defined area codes, 8 DSDs provide links to widely sharable external codes.

Table 1. Area dimensions and their codes[4]

DSD	dimension	generic dimension	range class	generic range class	code	alternate code
a	local		local	local	local	external
b.1	local	sd:refArea	local		local	dbpedia
b.2	local		local		local	dbpedia
c	local		external	WGS84	external	
d	local	sd:refArea	local		local	dbpedia
e	local	sd:refArea	local		local	geonames, dbpedia
f.1	local	sd:refArea	local		local	
f.2	local	sd:refArea	local		local	dbpedia
g	sd:refArea				external	external
h	sd:refArea				local	geonames, dbpedia, eurostat
i.1	sd:refArea				local	geonames, dbpedia, eurostat
i.2	local				local	geonames, dbpedia, eurostat

3.2 Time Dimension

Table.2 summarizes information about the time dimensions in a way similar to Table.1. As for the "dimension" and "generic dimension" columns, 6 DSDs in 12 ones use local dimension properties and referring to the upper-level "sd:refPeriod" (sdmx-dimension: refPeriod). 3 DSDs directly use this upper-level dimension property. Hence, for three-quarters of DSDs, the time dimension is identifiable in a schema level, but three of them have the same problem as in WBStat in Section 2. For the

[4] The homepage URLs for the endpoints surveyed here are as follows:

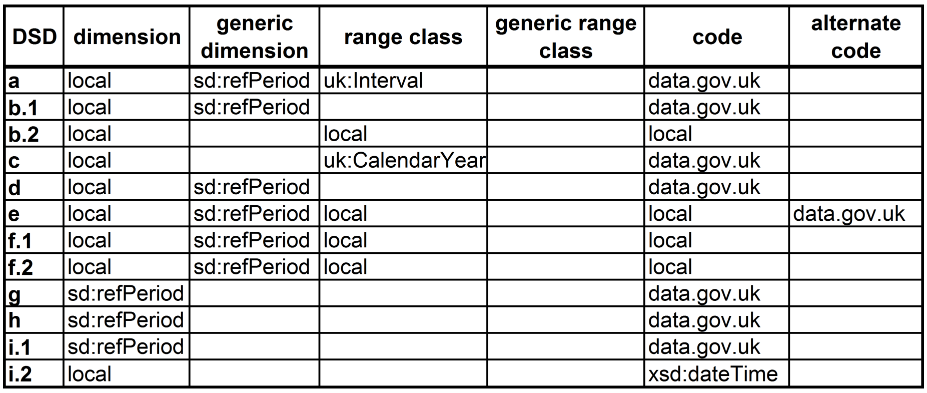
a: Consumption data (Scotland), `http://cofog01.data.scotland.gov.uk/`
b: ECB (European Central Bank) Linked Data, `http://ecb.270a.info/`
c: Environment Agency, Bathing water quality, `http://environment.data.gov.uk`
d: FAO Linked Data, `http://fao.270a.info/`
e: ISTAT Immigration, `http://www.linkedopendata.it/datasets/istat-immigration`
f: OECD Linked Data, `http://oecd.270a.info/`
g: Open Data Communities, `http://opendatacommunities.org/`
h: Transparence International Linked Data, `http://transparency.270a.info/`
i: World Bank Linked Data, `http://worldbank.270a.info/`

range classes, two sites labeled as "a" and "c" specifies widely known code classes. Herein, "uk:Interval" and "uk:CalenderYear" are abbreviations of "interval:Interval" and "interval:Calender Year" defined by data.gov.uk.

Since time concepts are common in the world, the difference among years, quarters, months and so on can be specified by a range class when using codes defined by data.gov.uk. Generic code classes for time codes may be unnecessary.

Next, let us examine instance codes. "data.gov.uk" in the "code" and "alternate code" columns represents the time codes defined by data.gov.uk. Two-thirds sites adopt this code system directly or via skos:exactMatch. Nevertheless, it is not confirmed in the schema level except the above two sites, because the range classes are not specified. It is desirable to define the range class properly.

Table 2. Time dimensions and their codes



DSD	dimension	generic dimension	range class	generic range class	code	alternate code
a	local	sd:refPeriod	uk:Interval		data.gov.uk	
b.1	local	sd:refPeriod			data.gov.uk	
b.2	local		local		local	
c	local		uk:CalendarYear		data.gov.uk	
d	local	sd:refPeriod			data.gov.uk	
e	local	sd:refPeriod	local		local	data.gov.uk
f.1	local	sd:refPeriod	local		local	
f.2	local	sd:refPeriod	local		local	
g	sd:refPeriod				data.gov.uk	
h	sd:refPeriod				data.gov.uk	
i.1	sd:refPeriod				data.gov.uk	
i.2	local				xsd:dateTime	

4 Proposed Patterns of Dimension Description

4.1 Outline of Proposed Patterns

As explained in Section 2, we can find dimension candidates to match if the upper-level resources are declared for each dimension and its range class. In addition, instance matching can be done rather automatically if there exists schema-level information about external code links. In this section, we consider patterns, according to which one can define a dimension as having information about upper-level resources and external linkages.

From the trial in Section 2, we found that there is a case where it is difficult to add schema-level information. It is direct use of an abstract upper-level dimension property, on which it is impossible to add local information. We call this problem case (1). A similar problem was found in the survey in Section 3. It is direct use of an external code class, on which it is impossible to add local information, too. We call this problem case (2).

To resolve these problems, we present two patterns of dimension description. The first one uses a local code class and another one uses an external code class. The pattern with a local code class is much the same as the example found in the QB draft. The problem of case (1) above can be resolved by rewriting it according to this pattern. As for the pattern with an external code class, we propose an adapter class that makes an external code class coexist with local description. The problem of case (2) above can be resolved by applying this pattern.

Our survey in Section 3 shows that three DSDs among twelve ones correspond to case (1) and two area dimensions and eight time dimensions correspond to case (2).

4.2 Pattern of Dimension with Local Codes

The pattern of dimension with local codes is illustrated in Fig.2. Solid lines show the definition of area dimension found in the example in the QB draft. A boldfaced string in parentheses "( )" under a resource name indicates a resource type as in the pattern. Broken lines express our additions.

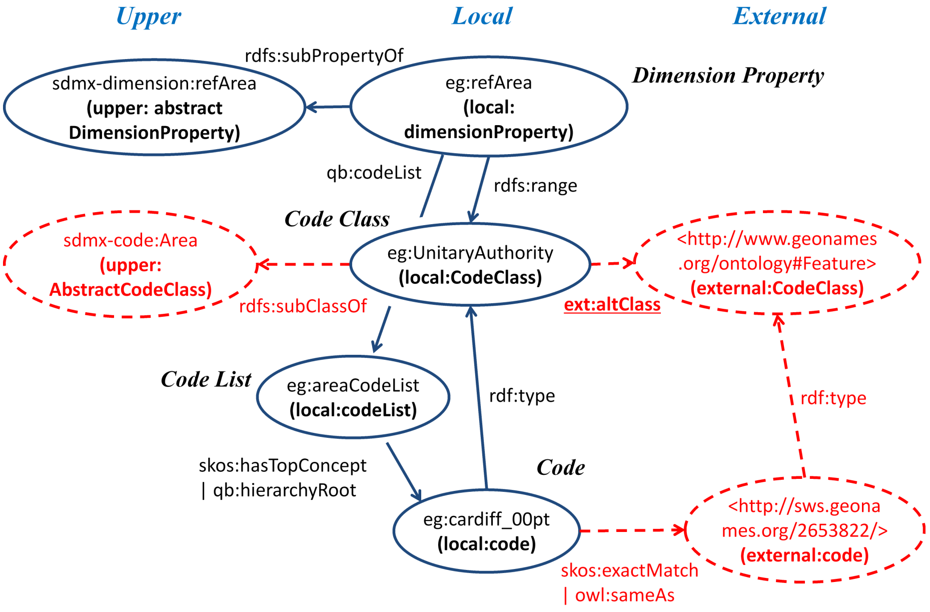

Upper
Local
External
rdfs:subPropertyOf
sdmx-dimension:refArea (upper: abstract DimensionProperty)
eg:refArea (local: dimensionProperty)
Dimension Property
qb:codeList
rdfs:range
Code Class
sdmx-code:Area (upper: AbstractCodeClass)
eg:UnitaryAuthority (local:CodeClass)
<http://www.geonames.org/ontology#Feature> (external:CodeClass)
rdfs:subClassOf
ext:altClass
Code List
eg:areaCodeList (local:codeList)
rdf:type
rdf:type
skos:hasTopConcept | qb:hierarchyRoot
Code
eg:cardiff_00pt (local:code)
<http://sws.geonames.org/2653822/> (external:code)
skos:exactMatch | owl:sameAs

Fig. 2. Pattern of dimension with local codes

Though the example in this figure basically comes from the QB draft, it is slightly modified. The original range class "admingeo:UnitaryAuthority" of the dimension "eg:refArea" is not an SKOS class locally defined. However, as for area dimensions, a large majority of actual statistical linked data adopts a locally defined code class as shown in Table 1 in Section 3. Hence, we regard the code class of the area dimension here as locally defined. To avoid confusion, the prefix is changed from "admingeo:"

to “eg:”. When using original “admingeo:UnitaryAuthority”, the pattern in the next subsection must be applied.

This pattern has three parts such as “local”, “external” and “upper” ones as shown in Fig.2.[5]

Local components. "local:dimensionProperty" is a dimension property used for the concerned dataset and the range of it is "local:CodeClass". "local:CodeClass" is a concrete class of which instances can be enumerated, i.e. the code list corresponding to it, "local:codeList", can be defined. An instance of the code class is "local:code", i.e. a concrete code. All the components here are defined locally.

External components. When a "local:code" corresponds to a sharable external code ("external:code"), the correspondence is written by using either skos:exactMatch or owl:sameAs. A candidate of such an external code is a code defined by Geonames or DBPedia for area and an interval resource by data.gov.uk for time. The class of such external codes is "external:CodeClass".

There is no general way to write the correspondence of "local:CodeClass" to "external:CodeClass". Therefore we have to search code instances in order to verify whether an "external:code" is used or not. To improve this, we introduce a new predicate "ext:altClass" which maps a "local:CodeClass" to an "external:CodeClass". Herein, "ext:" is a tentative prefix indicating an extension and "altClass" is an abbreviation of "alternate class".[6]

Upper components. "upper:abstractDimensionProperty" represents an uppermost super-property of a "local:dimensionProperty". A typical example of it is sdmx-dimension:refArea. This aims to clarify what is a concrete dimension, e.g. eg:refArea is an area dimension. "upper:AbstractCodeClass" represents an uppermost superclass of "local:CodeClass". A typical example of it is "sdmx-code:Area". This clarifies what is a concrete class, e.g. eg:UnitaryAuthority is an area code class. These upper-level components would correspond to upper ontology for aligning heterogeneous ontologies mentioned in Section 1.

4.3 Pattern of Dimension with External Codes

The pattern of dimension with external codes is illustrated in Fig.3. As for linked data, it is desirable to use sharable external resources as far as possible. This pattern is pro-

[5] Similar classification of resources is found in [7]. They classify resources into three categories, such as “local”, “external” and what’s under “sdmx”. In our case, “upper” is not the same as “sdmx”. Only upper-level resources in the SDMX-RDF vocabulary are used as examples of “upper” resources.

[6] We can describe external link information on a VoID RDF file. In fact, both ItImmStat and WBStat in Section 2 provide their respective VoID files. However, we cannot identify which dimension uses the external class written in the file as an alternate code class.

vided for this case by modifying the previous pattern. In Fig.3, the time dimension found in the QB example is shown as an example.

Since an external code class cannot be modified locally and it may not be an SKOS class, it is impossible to link from "external:CodeClass" to "upper:AbstractCodeClass" directly. For this reason, we propose to introduce "local:CodeClassAdapter" as in Fig.3. This is a kind of an adapter (a wrapper) of "external:CodeClass". This is linked to "external:CodeClass" by using owl:equivalentClass. owl:equivalentClass states that the two classes have the same class extension but are not the same class [8]. Hence "local:CodeClassAdapter" is regarded as a class having the same instances as in "external:CodeClass" and can have properties different from those of "external:CodeClass". Hence, it is possible to make it as looking like "local:CodeClass" in Fig.2. This type of adapter is often used in an object-oriented domain when using components defined externally [9]. A similar adapter concept is introduced in [10] in a semantic web domain.

When using this type of adapter, "local:CodeClassAdapter" becomes the range of "local:dimensionProperty", moving the object of the rdfs:range predicate from "external:CodeClass" to "local:CodeClassAdapter" as in Fig.3. Now, we can add an rdfs:subClassOf link from "local:CodeClassAdapter" to "upper:AbstractCodeClass". We can also add other annotations such as a comment to this type of code class. In addition, we can declare an alternate class as seen in Fig.2 if necessary.

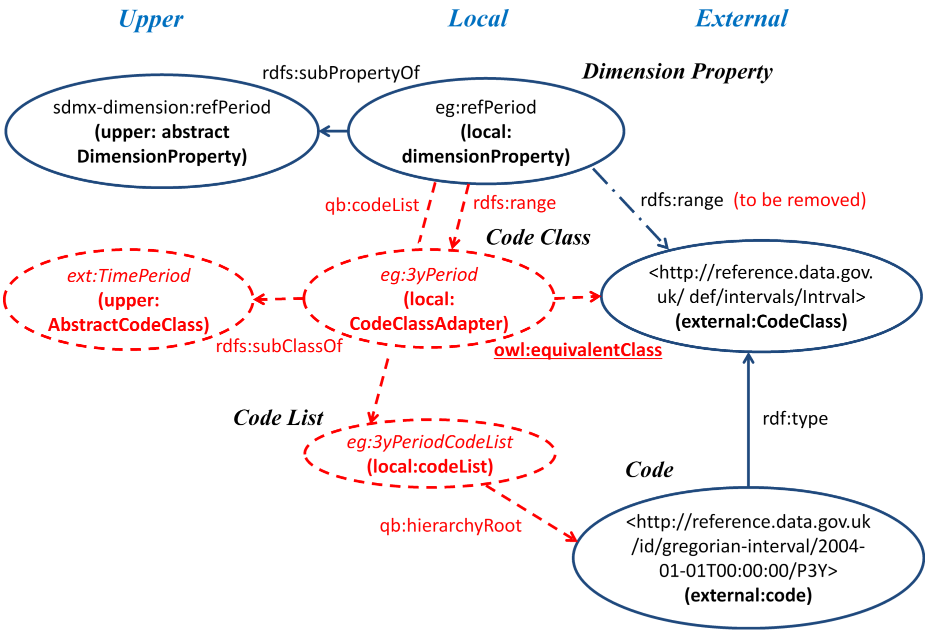


Fig. 3. Pattern of dimension with external codes

4.4 Essence of the Patterns

Specifying an upper concept of a dimension. As seen in Section 2, it becomes easy to find dimensions to match when the super properties of the dimensions are specified as upper-level resources, e.g. sdmx-dimension:refArea or sdmx-dimension:refPeriod. In addition, for a dimension property of an actual dataset, it is not good to use directly such an abstract upper-level dimension property. It is because neither its range nor its code list is available. It is better to define a local resource for the dimension property and to link it to an abstract upper-level one.

Specifying a code class. A dimension property is a mapping that indicates the role of its value played at the observation data. Hence, in a rigorous sense, it is not suitable for judging possibility of matching. The range of a dimension must be used for this purpose. It is a class gathering values of the dimension. A necessary condition for two dimensions to match is that the intersection of the ranges of them is not null. Therefore, if a code class, which is the range of a dimension, is specified in detail, it becomes precise to discover a dimension for matching. For example, the range of the time dimension in the QB example is "interval:Interval". "interval:Interval" is much abstract. It is because the example uses a special time interval of three years. But more specific time intervals are defined by data.gov.uk, such as interval:Year, interval:Quarter and interval:Month. Ordinary statistical datasets can employ such a specific time interval in order to declare its range in a precise manner.

Specifying an alternate code class. Even though dimensions to match are found, instance matching is impossible unless the dimension values are ensured to be the same. In case of linked data provided independently of each other, it may be rare that the same code system is employed among them. When external codes are available, it is important to declare their class as an alternate class. It will help us to discover dimensions to match in a schema level.

5 Conclusion and Future Works

We carried out a trial matching between statistical data from different sources. From this trial, we found that it would become easy to check the matching possibility if the appropriate upper-level resources were referred to and if the usage of external codes were identified in the schema-level. Next, we surveyed existing statistical linked data sites, examining their dimension descriptions. The result shows that it is impossible for many sites to add the information about the upper-level resources and the external linkages on their dimension descriptions. These dimension descriptions use external resources directly, so that local information cannot be added to them.

We proposed two patterns of dimension description to resolve this problem. The first pattern is formulated based on the QB example. For a dimension that uses an external dimension property directly, it is desirable to rewrite according to this pattern. The second pattern is for a case where external codes are used directly for dimension values. This pattern introduces an adapter code class, which enables an external code class to coexist with local description.

At present, only small numbers of upper-level resources are available, so that the benefit of the patterns proposed here is limited. However, through this research, we found that upper-level resources are useful in discovering/matching statistics. We also found that, for an upper concept of a dimension property, it is important to identify the roles of their code values, e.g. difference between place-of-birth and place-of-residence. For an upper concept of a code class, it is important to identify the type of their code values, e.g. country, domestic-administrative-area, river-basin and so on.

We also confirmed that upper-level resources including the above contents could be defined by using the structure of the SDMX-RDF vocabulary. We conclude tentatively that it is desirable to enrich upper-level resources under the SDMX-RDF structure. In this direction, we started a research work in which we shall extract upper concepts from major socio-economic statistics and organize them. The parts of the patterns concerning upper concepts are preparatory for utilization of rich upper concepts on statistical data.

Acknowledgements. We wish to thank the anonymous reviewers for their valuable and constructive comments that helped us to improve this paper.

References

1. Research Report on Improving Infrastructure for XML Usage on Statistical Information (in Japanese). Ministry of Economy, Trade and Industry, Japanese Government (2010)
2. Bizer, C., Heath, T., Berners-Lee, T.: Linked Data -- The Story So Far. In: International Journal on Semantic Web & Information Systems, Vol.5 Issue 3, pp.1-22. (2009)
3. Cyganiak, R., Reynolds, D. (eds.): The RDF Data Cube Vocabulary - W3C Working Draft 12 March 2013. `http://www.w3.org/TR/vocab-data-cube/` (2013)
4. Euzenat, J., Shvaiko, P.: Ontology Matching. Springer (2010)
5. Sato, H.: Statistical Data Models: from a Statistical Table to a Conceptual Approach. In: Michalewicz, Z. (ed.): Statistical and Scientific Databases, pp.167-200. Ellis Horwood (1991)
6. SDMX Content-Oriented Guidelines Annex 1: Cross-Domain Concepts, `http://sdmx.org/wp-content/uploads/2009/01/01_sdmx_cog_annex_1_cdc_2009.pdf` (2009)
7. Capadisli, S., Auer, S., Ngomo, A. N.: Linked SDMX Data, `http://csarven.ca/linked-sdmx-data#linked-sdmx-concept-links` (2013)
8. Dean, M., Schreiber, G. (eds.): OWL Web Ontology Language Reference, W3C Recommendation 10 February 2004, `http://www.w3.org/TR/owl-ref/` (2004)
9. Gamma, E., Helm, R., Johnson, R., Vlissides, J.: Design Patterns: Elements of Reusable Object-Oriented Software, Addison-Wesley (1995)
10. Katasonov, A.: Ontology-Driven Software Engineering: Beyond Model Checking and Transformations. In: International Journal of Semantic Computing, Vol. 6. No: 2, pp.205-242. (2012)